Incorporating Behavioral Recommendations Mined from Event Logs into AI Planning

Gyunam Park[1] ✉, Majid Rafiei[1],
Hayyan Helal[2], Gerhard Lakemeyer[2], and Wil M.P. van der Aalst[1]

[1] Process and Data Science Group, RWTH Aachen University, Aachen, Germany
[2] Knowledge-Based Systems Group, RWTH Aachen University, Aachen, Germany

Abstract. AI planning plays a crucial role in the design and optimization of business processes, providing optimal plans, i.e., sequence of activities, based on manually crafted or formally documented rules. When these plans are executed in business processes, the supporting information systems record a wealth of event data. Analyzing such event data facilitates understanding implicit patterns and recommendations that have the potential to refine planning strategies significantly. In this paper, we introduce a systematic approach to mining these recommendations from event data and integrating them into AI planning, thus creating plans that are informed by both the regulatory hard rules and the flexibility of soft recommendations.

Keywords: AI Planning · Automated Planning · Process Mining · Behavioral Recommendations.

1 Introduction

AI planning has been an essential tool in designing operational processes across various domains, primarily using rules derived from process models, regulations, and domain-specific knowledge [13]. In the domain of logistics, for instance, Fox et al. [7] have shown how planning algorithms can help optimize delivery routes and schedules, maximizing efficiency. The growing relevance of AI planning in healthcare has also been underlined. For example, Myers et al. [14] showcased how planning can be employed to design patient-specific treatment pathways. In the educational domain, AI planning has emerged as a tool for curricula design, helping in the sequencing of courses, lesson plans, and learning modules to adapt to students' varying capabilities and needs [5].

In real-life scenarios, rules used in AI planning can be broadly categorized into two distinct types: *hard rules* and *soft rules*. The former are non-negotiable regulations or immutable prerequisites, while the latter are more flexible, often shaped by historical trends and past experiences. Taking curricular designs in the education sector as an illustrative example, having "Data Structures" as a mandatory prerequisite for "Advanced Data Structures" exemplifies a hard rule. At the same time, the observation that students who first engage with "Algorithms" tend to excel more in "Advanced Data Structures" exemplifies a soft rule.

In dynamically changing domains such as education, logistics, etc., the efficiency of planning mechanisms can be significantly improved by integrating hard rules with soft rules. Such a fusion not only accommodates the structured requirements of planning but also allows for adjustments based on evolving insights and emergent patterns.

In this paper, we introduce a novel, two-phase approach to augment AI planning through behavioral recommendations as soft rules. The first phase aims to extract behavioral recommendations from event data that record the execution of plans. Using the *Declare* framework [15], we define pattern templates encompassing various behavioral attributes like precedence, response, and others. Subsequently, pattern candidates are instantiated for each template. By classifying cases that either align with or deviate from these patterns, we can statistically test the significance of observed behavioral patterns. Subsequently, the patterns that successfully meet this evaluation are suggested, whereas the remaining patterns are disregarded.

In the second phase, we incorporate these behavioral recommendations into the AI planning paradigm. Here, recommendation-based AI planning, also known as preference elicitation AI planning [4,12] emerges as a key technique. Classical AI planning focuses primarily on finding a plan that satisfies a set of hard constraints or goals. In contrast, preference-based planning recognizes that in many real-world scenarios, not all goals are equally preferred, thus introducing a notion of soft constraints or preferences. Regulations, in our context, act as these non-negotiable, hard constraints — they set the boundaries within which any plan must operate. In contrast, the recommendations derived from behavioral patterns function as soft guidelines or preferences. They guide the planning process towards solutions that have historically been beneficial but do not strictly bind the plan.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 lays the foundational concepts necessary for understanding our approach, covering event data, the Declare framework, and the principles of AI planning. Section 4 details our proposed method for mining behavioral recommendations from event data. We then, in Section 5, describe the integration of these recommendations into AI planning, elaborating how they inform the generation of optimal plans. Finally, Section 6 concludes this paper.

2 Related Work

Classical AI planning typically involves formalizing a problem using states, actions, and goals, and then using algorithms to find a sequence of actions (plan) that achieves the specified goals. One of the classical algorithms is the STRIPS (Stanford Research Institute Problem Solver) planning formalism [6].

Preference-based AI planning has found extensive use in generating personalized recommendations and plans. One example of this is demonstrated in [18], where the authors combine Hierarchical Task Networks (HTNs) with user preferences to generate preferred plans. They extend the Planning Domain Definition Language (PDDL3) to allow for the specification of preferences over HTN constructs. Another instance of this approach can be seen in the work by Li et al. [11], where a temporal HTN planner is proposed to handle temporal constraints with preferences. This planner employs Simple Temporal Networks with Preferences (STNP) to represent temporal preferences and extends operators and methods for expressing temporal preferences within planning domain knowledge. Finally, Bienvenu et al. [2] provides a similar approach to ours, where classical AI planning with a bounded plan length is enhanced by temporal preferences in LTL_f.

However, specifying preferences in advance can be challenging and time-consuming, as user preferences may be complex, unknown, or incomplete. Consequently, preference elicitation (recommendation-based) frameworks for automated planning have gained increased attention. In [12], a preference elicitation framework for automated planning is presented. This framework facilitates user interaction through a restricted set of uncomplicated comparative queries, allowing for subsequent learning of a preference relation predictor based on the user's feedback.

In the broader context of data-driven approaches for study planning, which is the specific focus area of our work as an application of our approach, a systematic literature review in [20] highlights the prevalence of both "knowledge-base" and "machine-learning-based" methods for generating rules and recommendations in education. Various techniques have been proposed, including sequential pattern mining [1], statistical methods [16], and advanced machine learning techniques [9,19,3].

In the specific context of using event data to extract rules and recommendations for study planning, [17] represents a recent and highly relevant work. In this study, the authors extract a wide range of features from event data collected by a campus management system. They then employ decision tree models trained on these features to discover goal-based recommendations for study planning.

3 Preliminaries

AI Planning AI planning is a fundamental domain within artificial intelligence that focuses on the automatic generation of sequences of actions to achieve specific goals, given a description of the initial state and a set of possible actions. Core concepts of AI Planning include the following.

- *State* (s) represents the configuration of the world at a given time point.
- *Action* (a) refers to an operation capable of transitioning the world from one state to another. In this work, we consider actions in planning and activities in events as analogous concepts: actions are entities that are planned, and their corresponding executions are recorded as events.
- *Preconditions* ($\mathrm{Pre}(a)$) describe a set of states, at which an action is executable.
- *Effects* ($\mathrm{Eff}(a)$) are results from executing an action, altering the state.
- *Plan* (π): An ordered sequence of actions a_0,...,a_k, where, beginning from an initial state s_0, a_0 is executable at s_0 and results in s_1, a_1 is executable at s_1 and results in s_2, and so on, such that that, a goal state s_g results from applying a_k to s_{k-1}.

Imagine designing a computer science curriculum while ensuring hard rules, i.e., prerequisites, are met.

- A *state* might represent the completion status of the "Data Structures" course.
- An *action* could be taking a course, such as "Advanced Data Structures".
- A course like "Advanced Data Structures" would have the *precondition* that "Data Structures" is already completed.

Table 1: An Example of Event Logs in an Educational Context

Case ID	Event ID	Activity	Timestamp	Course Grade
student1	e1	Data Structures (DS)	2023-02-01	1.3
student1	e2	Advanced Data Structures (ADS)	2023-06-15	2.0
student1	e3	Algorithms (A)	2023-07-20	1.7
student2	e4	Data Structures (DS)	2023-02-01	2.6
student2	e5	Advanced Data Structures (ADS)	2023-06-15	3.2
student2	e6	Algorithms (A)	2023-07-20	2.9

- The *effects* of introducing and completing "Data Structures" would be equipping students for more advanced courses.
- A *plan* in this scenario is a sequence of courses ensuring the prerequisites.

Given these concepts, AI planning can be formally presented as:

Definition 1 (AI Planning). *Let $\mathcal{S}$ be the universe of all possible states and $\mathcal{A}$ the universe of all possible actions. A planning domain $D=(A,S)$, where $A\subseteq\mathcal{A}$ is a set of actions and $S\subseteq\mathcal{S}$ is a set of states. A planning problem within domain D is denoted as $P_D=(s_0,G)$ where $s_0\in S$ is an initial state and $G\subseteq S$ is a set of potential goal states. The objective of AI planning is to discover a plan π corresponding to problem P_D in domain D. This plan, when executed from s_0, should lead to a state s' such that $s'\in G$.*

Event Logs If the plans are executed in operational processes, the executions are recorded as *event logs*. We use event logs to mine behavioral recommendations. Each event refers to an action (i.e., activity) in the plan that has occurred. Additionally, these events can possess diverse attributes such as a timestamp, a particular person as the activity performer, and associated costs. Table 1 represents an event log in an E-learning context. The event log contains two cases: `student1` and `student2`. The first row represents an event, $e1$, belonging to `student1`, which describes `student1`' finishing `Data Structure` course on 2023-02-01 with a grade of 1.3.

The Declare Framework In this work, we mine behavioral recommendations from event logs. The mined behavioral recommendations are formally represented using temporal pattern templates in *Declare* [15], a declarative language designed for process modeling and analysis. *Declare* is equipped with a set of temporal pattern templates that have been inspired by a catalog of temporal logic patterns used in model checking for a variety of dynamic systems from different application domains. Each temporal pattern template represents a distinct temporal relationship, and temporal patterns are derivations of these templates corresponding to specific activities.

For the complete set of pattern templates in Declare, we refer readers to [15]. Frequent pattern templates in Declare include:

- **Response:** Upon the occurrence of activity a_i, activity a_j must eventually occur, denoted as Response(a_i,a_j).

- **Precedence:** Activity a_j can occur only if activity a_i has occurred beforehand, denoted as Precedence(a_i,a_j).
- **Exclusive Choice:** If activity a_i occurs, activity a_j must not occur, and vice versa, denoted as ExclusiveChoice(a_i,a_j).

4 Phase 1: Mining Behavioral Recommendations

Figure 1 provides an overview of mining behavioral recommendations from event data. Using the temporal pattern templates of the Declare framework, such as *precedence*, *response*, etc., we instantiate temporal pattern candidates. For each pattern candidate, we conduct *LTL checking* on all the cases of a given event log and classify them based on their alignment or deviation from the pattern. Next, we conduct *statistical testing* to ascertain the significant difference between the satisfied cases and violated cases, i.e., the validity of the temporal pattern. Only patterns that show a significant difference between satisfied cases and violated cases are recommended; the rest are discarded.

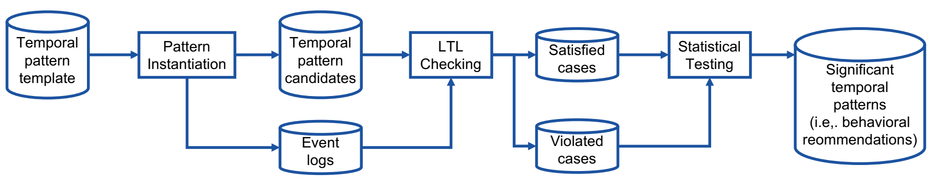

Fig. 1: Overview of Mining Behavioral Recommendations

Pattern Instantiation First, the pattern instantiation function is designed to generate pattern candidates based on a template. For instance, given the *Precedence* template and a set of activities, e.g., {`Data Structures (DS)`,`Advanced Data Structures (ADS)`,`Algorithms (A)`}, it would yield six pattern candidates such as {*Precedence*(`DS`,`ADS`),...}.

LTL Checking Next, the LTL (Linear Temporal Logic) checking function is essential for determining whether a specific event sequence (or trace) adheres to or violates a given pattern. For instance, for the student `student1` in Table 1 with a course-taking sequence, i.e., ⟨`DS`,`ADS`,`A`⟩, this function could evaluate whether they followed the *Precedence*(`DS`,`ADS`) pattern.

The LTL checking function can be implemented in many ways [8,10]. Giannakopoulou and Lerda [8] translate LTL formulae into Büchi automata, which allows for the efficient checking of event logs against temporal properties. The automata-based checking leverages state exploration methods to systematically verify adherence or violation of LTL-specified patterns within event traces. For more detailed explanations of this technique and its application to the analysis of event logs, the reader is referred to [8].

Statistical Testing Finally, the statistical testing function aims to assess the relative superiority of cases satisfying a specified pattern, i.e., $C_{satisfied}$, over those that violate it, i.e., $C_{violated}$. The distribution for each set, i.e., $C_{satisfied}$ and $C_{violated}$, is formed using a scoring function $score \in C_{satisfied} \cup C_{violated} \rightarrow \mathbb{R}$ that assigns values to individual cases based on particular characteristics or outcomes. This value could stem from various sources. One might consider the case's inherent attributes, such as the overall GPA of a student. Alternatively, one could focus on specific event attributes within a case, like a student's grade in the `Advanced Data Structures` course.

By comparing distributions derived from both sets of cases, the function determines the statistical significance of any observed differences. For instance, it can assess whether there is a significant difference in grades between students who adhere to and those who break from the $Precedence$(`DS`,`ADS`) pattern. Patterns securing a p-value below a predefined significance level (e.g., $p\_value < \alpha$) and satisfying the condition of the mean of $C_{satisfied}$ being at least a predefined difference level lower than $C_{violated}$ are recommended. Patterns that do not meet these criteria are considered not to have a significant or meaningful difference and are thus discarded.

5 Phase 2: Planning Based on Behavioral Recommendations

Figure 2 shows an overview of integrating behavioral recommendations into AI planning to generate optimal plans. Initially, conventional planning problems are taken and transformed by incorporating behavioral recommendations, which leads to the creation of *recommendation-based planning problems*. These enhancements include assigning weights to specific temporal patterns within the planning problems effectively prioritizing certain actions over others based on the provided recommendations. The process continues by merging these enhanced problems with defined planning domains, which include hard rules that the final plan must adhere to. Next, a recommendation-based AI planning system is employed to devise plans that are not only valid in terms of domain constraints but also optimized according to the incorporated recommendations. The final output is an optimal plan that maximizes the sum of the weights by respecting the recommended temporal patterns, ensuring that the plan is both feasible and closely aligned with the preferred behaviors identified by the recommendations.

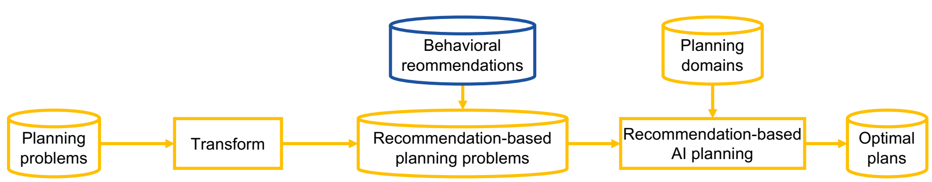

Fig. 2: Overview of Planning Based on Behavioral Recommendations

Transformation First, based on behavioral recommendations, we calibrate weights to align the planning domain with observed patterns. Action pairs resonating with these recommendations can be assigned higher weights (or lesser costs), elevating their likelihood in the optimal plan. For example, if data suggests that the `Algorithms` course preceding the `Advanced Data Structures` course leads to improved student performance in the `Advanced Data Structures` course, i.e., the precedence temporal pattern of this pair has a high weight, then a plan where `Algorithms` is planned before `Advanced Data Structures` is considered more optimal and recommended to students more than a plan where this does not hold.

Definition 2 (Recommendation-Based Planning Problem). *Let R be a set of siginificant temporal patterns, i.e., behavioral recommendations, mined from the the previous phase, and let $D=(A,S)$ be a planning domain. Let P_D be the set of all plans in this planning domain, a weight function $w_R : P_D \rightarrow \mathbb{R}^+$ maps plans to their weights and is calculated by:*

$$w_R(\pi) = \sum_{\substack{T(a,b)\in R \ s.t. \\ T(a,b) \ holds \ in \ \pi}} w_T(a,b)$$

where $w_T(a,b) \in [0,1]$ is the weight of recommendation $T(a,b)$. For a planning problem $P_D=(s_0,G)$, a plan π is considered optimal if it is valid and maximizes the weight $w_R(\pi)$.

Recommendation-Based AI Planning Solving a planning problem where actions can occur maximally once is in NP because the length of any valid plan is linear. Therefore, it can be modeled as an ILP feasibility problem. For any plan π where actions occur maximally once, there exists a unique tuple of relations $(e_\pi, <_\pi)$, where $e_\pi \subseteq A$, s.t. $a \in e_\pi$ iff. a occurs in π, and a total order $<_\pi \subset A^2$, s.t. $a <_\pi b$ iff. a occurs before b in π. An ILP feasibility problem searches for such tuples corresponding to valid plans.

Minimal length, minimal costs, or maximal rewards usually define optimality for such plans. In our setting, optimality is defined by the function $w_R(\pi)$. Thus, the ILP optimality problem:

$$\text{maximize } w_R(\pi) \text{ s.t. } \pi \text{ is a valid plan}$$

can solve any recommendation-based AI planning problem, where actions occur maximally once.

6 Conclusion

In this paper, we have presented a novel, two-phase approach that enhances traditional AI planning with behavioral recommendations derived from event data. By leveraging the Declare framework to define and instantiate pattern templates, we have successfully extracted meaningful behavioral patterns that serve as soft rules in the planning process. These soft rules complement the hard constraints typically used in AI planning, providing a more adaptable planning mechanism that reflects both the rigid requirements and the flexible preferences observed in real-world scenarios.

References

1. Al-Twijri, M.I., Luna, J.M., Herrera, F., Ventura, S.: Course recommendation based on sequences: An evolutionary search of emerging sequential patterns. Cogn. Comput. **14**(4), 1474–1495 (2022)
2. Bienvenu, M., Fritz, C., McIlraith, S.A.: Planning with qualitative temporal preferences. KR **6**, 134–144 (2006)
3. Britto, J., Prabhu, S., Gawali, A., Jadhav, Y.: A machine learning based approach for recommending courses at graduate level. In: ICSSIT. pp. 117–121 (2019)
4. Das, M., Odom, P., Islam, M.R., Doppa, J.R., Roth, D., Natarajan, S.: Preference-guided planning: An active elicitation approach. In: André, E., Koenig, S., Dastani, M., Sukthankar, G. (eds.) AAMAS 2018. pp. 1921–1923 (2018)
5. Desmarais, M.C., de Baker, R.S.J.: A review of recent advances in learner and skill modeling in intelligent learning environments. User Model. User Adapt. Interact. **22**(1-2), 9–38 (2012)
6. Fikes, R., Nilsson, N.J.: STRIPS: A new approach to the application of theorem proving to problem solving. Artif. Intell. **2**(3/4), 189–208 (1971)
7. Fox, M., Long, D., Magazzeni, D.: Explainable planning (2017)
8. Giannakopoulou, D., Lerda, F.: From states to transitions: Improving translation of LTL formulae to Büchi automata. In: FORTE. p. 308–326 (2002)
9. Jiang, W., Pardos, Z.A., Wei, Q.: Goal-based course recommendation. In: LAK. pp. 36–45 (2019)
10. de Leoni, M., Marrella, A.: Aligning real process executions and prescriptive process models through automated planning. Expert Syst. Appl. **82**, 162–183 (2017)
11. Li, M., Wang, H., Qi, C., Zhou, C.: Handling temporal constraints with preferences in HTN planning for emergency decision-making. J. Intell. Fuzzy Syst. **30**(4), 1881–1891 (2016)
12. Mantik, S., Li, M., Porteous, J.: A preference elicitation framework for automated planning. Expert Syst. Appl. **208**, 118014 (2022)
13. Marrella, A.: Automated planning for business process management. J. Data Semant. **8**(2), 79–98 (2019)
14. Myers, K.L., Jarvis, P., Tyson, M., Wolverton, M.: A mixed-initiative framework for robust plan sketching. In: Giunchiglia, E., Muscettola, N., Nau, D.S. (eds.) ICAPS. pp. 256–266 (2003)
15. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: EDOC. pp. 287–300 (2007)
16. Polyzou, A., Nikolakopoulos, A.N., Karypis, G.: Scholars walk: A markov chain framework for course recommendation. In: Desmarais, M.C., Lynch, C.F., Merceron, A., Nkambou, R. (eds.) EDM (2019)
17. Rafiei, M., Bayrak, D., Pourbafrani, M., Park, G., Helal, H., Lakemeyer, G., van der Aalst, W.M.P.: Extracting rules from event data for study planning. CoRR **abs/2310.02735** (2023)
18. Sohrabi, S., Baier, J.A., McIlraith, S.A.: HTN planning with preferences. In: Boutilier, C. (ed.) IJCAI. pp. 1790–1797 (2009)
19. Tan, J., Chang, L., Liu, T., Zhao, X.: Attentional autoencoder for course recommendation in MOOC with course relevance. In: CyberC. pp. 190–196 (2020)
20. Urdaneta-Ponte, M.C., Mendez-Zorrilla, A., Oleagordia-Ruiz, I.: Recommendation systems for education: Systematic review. Electronics **10**(14), 1611 (2021)